



THE UNIVERSITY *of* EDINBURGH

Edinburgh Research Explorer

Complete Sequent Calculi for Induction and Infinite Descent

Citation for published version:

Simpson, A & Brotherston, J 2007, Complete Sequent Calculi for Induction and Infinite Descent. in *Logic in Computer Science, 2007. LICS 2007. 22nd Annual IEEE Symposium on*. Institute of Electrical and Electronics Engineers (IEEE), pp. 51-62. <https://doi.org/10.1109/LICS.2007.16>

Digital Object Identifier (DOI):

[10.1109/LICS.2007.16](https://doi.org/10.1109/LICS.2007.16)

Link:

[Link to publication record in Edinburgh Research Explorer](#)

Document Version:

Peer reviewed version

Published In:

Logic in Computer Science, 2007. LICS 2007. 22nd Annual IEEE Symposium on

General rights

Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.



Complete Sequent Calculi for Induction and Infinite Descent

James Brotherston*

Dept. of Computing
Imperial College, London, UK

Alex Simpson†

LFCS, School of Informatics,
University of Edinburgh, Scotland, UK

Abstract

This paper compares two different styles of reasoning with inductively defined predicates, each style being encapsulated by a corresponding sequent calculus proof system.

The first system supports traditional proof by induction, with induction rules formulated as sequent rules for introducing inductively defined predicates on the left of sequents. We show this system to be cut-free complete with respect to a natural class of Henkin models; the eliminability of cut follows as a corollary.

The second system uses infinite (non-well-founded) proofs to represent arguments by infinite descent. In this system, the left rules for inductively defined predicates are simple case-split rules, and an infinitary, global condition on proof trees is required to ensure soundness. We show this system to be cut-free complete with respect to standard models, and again infer the eliminability of cut.

The second infinitary system is unsuitable for formal reasoning. However, it has a natural restriction to proofs given by regular trees, i.e. to those proofs representable by finite graphs. This restricted “cyclic” system subsumes the first system for proof by induction. We conjecture that the two systems are in fact equivalent, i.e., that proof by induction is equivalent to regular proof by infinite descent.

1 Introduction

Many concepts in mathematics and computer science are most naturally formulated using inductive definitions. Thus proof support for inductive definitions is an essential component of proof assistants and theorem provers. Often, libraries are provided containing collections of useful induction principles associated with a given set of inductive definitions, see e.g. [16, 9, 18]. In other cases, mechanisms permitting “cyclic” proof arguments are used, with intricate conditions imposed to ensure soundness, see e.g. [23, 17, 8].

*Research undertaken while a PhD student at LFCS, School of Informatics, University of Edinburgh, supported by an EPSRC PhD studentship.

†Research supported by an EPSRC Advanced Research Fellowship.

In this paper we study and compare proof-theoretic foundations for these two quite different styles of reasoning.

In the case of classical first-order logic, Gentzen’s sequent calculus LK provides an elegant proof system that is well-suited to the goal-directed approach to proof construction employed in many proof assistants. Each logical constant is specified by two types of basic rule, introducing the constant on the left and on the right of sequents respectively. Gentzen’s well-known cut-elimination theorem implies that direct proofs, using these rules alone, are sufficient to derive any valid sequent. In addition to its theoretical elegance, this has implications for proof search, with the locally applicable proof rules thereby constrained by the logical constants appearing in the current goal.

In a previous paper by the first author [2], Gentzen’s LK was extended to obtain two proof systems for classical first-order logic with inductively defined predicates: (i) a system containing explicit proof rules embodying the standard induction principles; and (ii) a system without induction rules, but instead allowing circular arguments subject to soundness constraints. The main contributions of [2] were the formulation of the second “cyclic” proof system, and the demonstration that the second system subsumes the first (i.e., cyclic proof was shown to be at least as powerful as proof by explicit induction). The equivalence of the two systems was posed as a conjecture.

As the main contribution of the present paper, we establish completeness and cut-eliminability results for these two approaches to inductive reasoning. Aside from their intrinsic technical interest, our results demonstrate the calculi from [2] as being canonical ones embodying the two styles of reasoning. We hope that our results will help to stimulate wider interest in such systems.

In Section 3, we recall the system for induction from [2], here called LKID, in which the right-hand rules for inductively defined predicates simply reflect the closure conditions on the predicate, and the left-hand rules embody the natural induction principle for the predicate in question. A closely related precursor is Martin-Löf’s natural deduction system for intuitionistic logic with (iterated) inductive definitions [13], in which induction rules are included as

elimination rules for inductively defined predicates. As is well known, elimination rules in natural deduction serve the same purpose as left-introduction rules in sequent calculus. Nonetheless, it is only recently that sequent calculus counterparts of Martin-Löf’s system have been explicitly considered, by McDowell, Miller and Tiu [14, 21]. Ours is a simple classical analogue of these intuitionistic systems.

Our main new result about LKID is a completeness result, for cut-free proofs, relative to a natural class of “Henkin models” for inductive predicates (Theorem 3.6). The eliminability of cut in LKID follows as an immediate corollary. These results serve to endorse the naturalness of LKID: completeness shows that no proof principles are missing, and cut-eliminability vindicates the formulation of the proof rules. The latter result may come as a surprise to some, since there is a fairly popular misconception — possibly arising from the well-known need for generalisation in inductive proof (see e.g. [4]) — that cut-elimination is impossible in the presence of inductive definitions. However, readers familiar with [13, 14, 21] will not be surprised, since these papers contain analogous normalization/cut-elimination theorems for related intuitionistic systems. Their proofs, however, are based on Tait’s “computability” method, and do not readily adapt to our classical setting. Compared with such proofs, our semantic approach suffers from the weakness of not establishing that any particular cut-elimination strategy terminates, but, in compensation, it establishes a completeness result of independent interest.

Sections 4–6 concern the second approach to inductive proof based on circular reasoning. Following [23], it is natural to view this approach as a formalisation of proof by “infinite descent” *à la* Fermat. For natural numbers, infinite descent exploits the fact that, since there are no infinite strictly decreasing sequences of numbers, any case in a proof that furnishes such a sequence can be ignored as contradictory. This technique can be extended to general inductively defined predicates: any case of a proof which yields an infinite sequence of “unfoldings” of some inductively defined predicate can likewise be dismissed. In [2], this principle is implemented by replacing the induction rules of LKID with simple “case split” rules (which unfold inductively defined predicates on the left of sequents), and by allowing proofs to be cyclic graphs rather than finite trees. In general, such proof graphs are not sound. However, the *global trace condition* of [2] requires that, for a graph to qualify as a proof, every infinite path through the graph must enjoy the property that some inductively-defined predicate is unfolded infinitely often along the path. This condition guarantees soundness since, intuitively, it ensures that all such infinite paths can be disregarded from the proof for infinite descent reasons, leaving only finite ones behind.

The main results in [2] concerning the cyclic proof sys-

tem are: (i) it is sound (the proof is somewhat more technical than the informal argument above), and (ii) it subsumes proof by induction. Since one might well dream up many other proof systems enjoying these two properties, such results do not, by themselves, provide much evidence that the cyclic proof system is in any sense canonical.

In this paper, we take a different perspective. Rather than considering the cyclic system as basic, we instead present, in Section 4, an infinitary proof system LKID^ω , based on the same proof rules as the cyclic system, but in which proofs are possibly infinite (non-well-founded) trees of rule instances. As with the cyclic system, the global trace condition of [2] (which transfers verbatim to infinite trees) is imposed on trees to guarantee soundness (the soundness proof of [2] generalises immediately). The benefit of considering the infinitary system is borne out by the second main result of the paper: the system LKID^ω is complete relative to the usual “standard” models of inductively defined predicates (Theorem 4.9). Again, this completeness result holds for cut-free proofs, and so the eliminability of cut for LKID^ω follows. The proof of completeness is outlined in Section 5.

Having argued for the canonicity of the infinitary system LKID^ω , the cyclic system of [2] can be appreciated as arising in a very natural way: it is simply the restriction of LKID^ω to regular proof trees, i.e. to trees representable by finite (cyclic) graphs. Over such finite representations, the soundness condition is decidable (although completeness is necessarily lost), and hence the restricted proof system is suitable for formal reasoning (a property that clearly does not extend to LKID^ω). In fact, the soundness condition appears to subsume various heuristic conditions for cyclic proofs adopted in the theorem proving literature (see e.g. [23, 17, 8]).

In Section 6, we briefly consider the induced cyclic proof system, here called CLKID^ω , in order to reprise the conjecture from [2] that LKID and CLKID^ω are equivalent. The significance of this conjecture seems to be enhanced by the results of the present paper. Indeed, if one accepts that LKID and CLKID^ω canonically embody the two styles of reasoning, then the conjecture can be understood informally as asserting that proof by induction is equivalent to regular proof by infinite descent. We end the paper by stating this conjecture formally and commenting on the apparent difficulties its proof poses.

Due to space constraints, only outline proofs of our main results are included in this paper. Full proofs can be found in the first author’s recent PhD thesis [3].

2. Syntax and semantics of first-order logic with inductive definitions (FOL_{ID})

In this section we give the syntax and semantics of classical first-order logic with inductively defined predicates,

FOL_{ID}. Of the many possible definitional frameworks, we choose to work with ordinary (mutual) inductive definitions, specified by simple “productions” in the style of Martin-Löf [13]. This choice keeps the logic relatively simple, while including many important examples.

The languages we consider are the standard (countable) first-order languages, except that we designate finitely many of the predicate symbols of the language as *inductive*. A predicate symbol not designated as inductive is called *ordinary*. For the remainder of this paper we consider a fixed language Σ with inductive predicate symbols P_1, \dots, P_n . Terms of Σ are defined as usual; we write $t(x_1, \dots, x_n)$ for a term all of whose variables are contained in $\{x_1, \dots, x_n\}$.

The interpretation of the elements of Σ is as usual given by a first-order structure M with domain D ; we write X^M to denote the interpretation of the Σ -symbol X in M . Likewise, variables are interpreted as elements of D by an environment ρ ; we extend ρ to all terms of Σ in the standard way and write $\rho[x \mapsto d]$ for an environment defined exactly as ρ except that $\rho[x \mapsto d](x) = d$. The formulas of FOL_{ID} are the usual formulas of first-order logic with equality.¹ We then write $M \models_\rho F$ for the standard semantic satisfaction relation for formulas of FOL_{ID}.

Our proof systems will be interpreted relative to only those structures in which inductive predicates have their intended meanings, as specified by *definition sets* for the predicates, adapted from [13].

Definition 2.1 (Inductive definition set). An *inductive definition set* Φ for Σ is a finite set of *productions*, which are rules of the form:

$$\frac{Q_1(\mathbf{u}_1(\mathbf{x})) \dots Q_h(\mathbf{u}_h(\mathbf{x})) \ P_{j_1}(\mathbf{t}_1(\mathbf{x})) \dots P_{j_m}(\mathbf{t}_m(\mathbf{x}))}{P_i(\mathbf{t}(\mathbf{x}))} \text{ (Def)}$$

where $j_1, \dots, j_m, i \in \{1, \dots, n\}$, and Q_1, \dots, Q_h are ordinary predicate symbols, and the bold vector notation abbreviates sequences of terms.

Example 2.2. We define the predicates N, E and O via the productions:

$$\frac{}{N0} \quad \frac{Nx}{Nsx} \quad \frac{}{E0} \quad \frac{Ex}{Osx} \quad \frac{Ox}{Esx}$$

In structures in which all “numerals” $s^k 0$ for $k \geq 0$ are interpreted as distinct elements, the predicates N, E and O correspond to the properties of being a natural, even and odd number respectively.

From this point onwards we consider an arbitrary fixed inductive definition set Φ for Σ and, when we need to consider an arbitrary production in Φ , will always use the explicit format of (Def) above.

¹The inclusion of equality in the language is a minor departure from [2].

The standard interpretation of the inductive predicates (cf. [1]) is obtained as usual by considering prefixed points of a monotone operator constructed from the definition set Φ . For standard models, this least fixed point can be constructed in iterative *approximant* stages.

Definition 2.3 (Definition set operator). Let M with domain D be a first-order structure for Σ , and for each $i \in \{1, \dots, n\}$, let k_i be the arity of the inductive predicate symbol P_i . Then partition Φ into disjoint subsets $\Phi_1, \dots, \Phi_n \subseteq \Phi$ by:

$$\Phi_i = \left\{ \frac{u}{v} \in \Phi \mid P_i \text{ appears in } v \right\}$$

Let each rule set Φ_i be indexed by r with $1 \leq r \leq |\Phi_i|$, and for each rule $\Phi_{i,r}$, say of the form (Def) specified above, define $\varphi_{i,r} : (\mathcal{P}(D^{k_1}) \times \dots \times \mathcal{P}(D^{k_n})) \rightarrow \mathcal{P}(D^{k_i})$, where $\mathcal{P}(\cdot)$ is powerset, by:

$$\varphi_{i,r}(X_1, \dots, X_n) = \left\{ \mathbf{t}^M(\mathbf{x}) \mid Q_1^M \mathbf{u}_1^M(\mathbf{x}), \dots, Q_h^M \mathbf{u}_h^M(\mathbf{x}), \right. \\ \left. \mathbf{t}_1^M(\mathbf{x}) \in X_{j_1}, \dots, \mathbf{t}_m^M(\mathbf{x}) \in X_{j_m} \right\}$$

Then define the function φ_i for each $i \in \{1, \dots, n\}$ by $\varphi_i(X_1, \dots, X_n) = \bigcup_r \varphi_{i,r}(X_1, \dots, X_n)$, whence the *definition set operator* for Φ is the operator φ_Φ , with domain and codomain $\mathcal{P}(D^{k_1}) \times \dots \times \mathcal{P}(D^{k_n})$, defined by:

$$\varphi_\Phi(X_1, \dots, X_n) = (\varphi_1(X_1, \dots, X_n), \dots, \varphi_n(X_1, \dots, X_n))$$

In the definition below, we write π_i^n for the i th projection function $\pi_i^n(X_1, \dots, X_n) = X_i$, and we extend union to the corresponding pointwise operations on n -tuples of sets.

Definition 2.4 (Approximants). Let M with domain D be a first-order structure for Σ , and let φ_Φ be the definition set operator for Φ . Define an ordinal-indexed set $(\varphi_\Phi^\alpha \subseteq \mathcal{P}(D^{k_1}) \times \dots \times \mathcal{P}(D^{k_n}))_{\alpha \geq 0}$ by $\varphi_\Phi^\alpha = \bigcup_{\beta < \alpha} \varphi_\Phi(\varphi_\Phi^\beta)$ (note that this implies $\varphi_\Phi^0 = (\emptyset, \dots, \emptyset)$). Then the set $\pi_i^n(\varphi_\Phi^\alpha)$ is called the α^{th} *approximant* of P_i , written as P_i^α .

Definition 2.5 (Standard model). A first-order structure M is said to be a *standard model* for (Σ, Φ) if for all $i \in \{1, \dots, n\}$, $P_i^M = \bigcup_\alpha P_i^\alpha$.²

Definition 2.5 fixes a standard interpretation of the inductive predicates. However, we shall also be interested in non-standard *Henkin models* of FOL_{ID} in which the least fixed point of the operator for the inductive predicates is constructed with respect to a chosen class of sets of tuples over the domain of interpretation. This approach is based on an idea originally employed by Henkin who obtained completeness theorems for higher-order calculi by considering validity with respect to his more general notion of model [10]. Our application in Section 3 is similar.

²For the form of production considered, we have $\bigcup_\alpha P_i^\alpha = P_i^\omega$, i.e. the closure ordinal is at most ω . However, we shall never exploit this.

Definition 2.6 (Henkin class). Let M with domain D be a structure for Σ . A *Henkin class* for M is a family of sets $\mathcal{H} = \{H_k \mid k \in \mathbb{N}\}$, where for each $k \in \mathbb{N}$, $H_k \subseteq \mathcal{P}(D^k)$ and:

- (H1) $\{(d, d) \mid d \in D\} \in H_2$;
- (H2) if Q is an ordinary/inductive predicate symbol of arity k then $\{(d_1, \dots, d_k) \mid Q^M(d_1, \dots, d_k)\} \in H_k$;
- (H3) if $R \in H_{k+1}$ and $d \in D$ then $\{(d_1, \dots, d_k) \mid (d_1, \dots, d_k, d) \in R\} \in H_k$;
- (H4) if $R \in H_k$ and $t_1(x_1, \dots, x_m), \dots, t_k(x_1, \dots, x_m)$ are terms then $\{(d_1, \dots, d_m) \mid (t_1^M(d_1, \dots, d_m), \dots, t_k^M(d_1, \dots, d_m)) \in R\} \in H_m$;
- (H5) if $R \in H_k$ then $\overline{R} = D^k \setminus R \in H_k$;
- (H6) if $R_1, R_2 \in H_k$ then $R_1 \cap R_2 \in H_k$;
- (H7) if $R \in H_{k+1}$ then $\{(d_1, \dots, d_k) \mid \exists d. (d_1, \dots, d_k, d) \in R\} \in H_k$.

Lemma 2.7. If $\mathcal{H} = \{H_k \mid k \in \mathbb{N}\}$ is a Henkin class for a structure M , ρ is an environment for M , F is a formula of FOL_{ID} and x_1, \dots, x_k are distinct variables, then $\{(d_1, \dots, d_k) \mid M \models_{\rho[x_1 \mapsto d_1, \dots, x_k \mapsto d_k]} F\} \in H_k$.

Definition 2.8 (\mathcal{H} -point / prefixed point). Let M be a structure for Σ and let \mathcal{H} be a Henkin class for M . Also let k_i be the arity of the inductive predicate symbol P_i for each $i \in \{1, \dots, n\}$. Then (X_1, \dots, X_n) is said to be an \mathcal{H} -point if $X_i \in H_{k_i}$ for each $i \in \{1, \dots, n\}$. and is said to be a *prefixed point* of the monotone operator φ_Φ if $\varphi_\Phi(X_1, \dots, X_n) \subseteq (X_1, \dots, X_n)$.

Lemma 2.9. Let \mathcal{H} be a Henkin class for a Σ -structure M . Then if (X_1, \dots, X_n) is an \mathcal{H} -point then so is $\varphi_\Phi(X_1, \dots, X_n)$.

Definition 2.10 (Henkin model). Let M be a first-order structure for Σ and \mathcal{H} be a Henkin class for M . (M, \mathcal{H}) is said to be a *Henkin model* for (Σ, Φ) if there exists a least prefixed \mathcal{H} -point $\mu_{\mathcal{H}}.\varphi_\Phi$ of φ_Φ , and for each $i \in \{1, \dots, n\}$, $P_i^M = \pi_i^n.\mu_{\mathcal{H}}.\varphi_\Phi$.

It is a standard result for inductive definitions that the least prefixed point of φ_Φ in $\mathcal{P}(D^{k_1}) \times \dots \times \mathcal{P}(D^{k_n})$ is given by the union of the approximants of φ_Φ , $\bigcup_\alpha \varphi_\Phi^\alpha$. Thus a standard model is a Henkin model (with $H_k = \mathcal{P}(D^k)$).

3. LKID: a proof system for induction

We use sequents of the form $\Gamma \vdash \Delta$, where Γ, Δ are finite sets of formulas, and use the notation $\Gamma[\theta]$ to mean that the substitution θ of terms for free variables is applied to all formulas in Γ . We use the standard sequent calculus rules

as given in many sources (see e.g. [7, 5]), together with the following rules for explicit substitution and equality, cf. [6].

$$\frac{\Gamma \vdash \Delta}{\Gamma[\theta] \vdash \Delta[\theta]} (\text{Subst}) \quad \frac{}{\Gamma \vdash t = t, \Delta} (=R)$$

$$\frac{\Gamma[u/x, t/y] \vdash \Delta[u/x, t/y]}{\Gamma[t/x, u/y], t = u \vdash \Delta[t/x, u/y]} (=L)$$

To these rules we add rules for introducing inductive predicates on the left and right of sequents.

First, for each production $\Phi_{i,r} \in \Phi$, say (Def), there is a sequent calculus right introduction rule for P_i :

$$\frac{\Gamma \vdash Q_1 \mathbf{u}_1(\mathbf{u}), \Delta \quad \dots \quad \Gamma \vdash Q_h \mathbf{u}_h(\mathbf{u}), \Delta \quad \Gamma \vdash P_{j_1} \mathbf{t}_1(\mathbf{u}), \Delta \quad \dots \quad \Gamma \vdash P_{j_m} \mathbf{t}_m(\mathbf{u}), \Delta}{\Gamma \vdash P_i \mathbf{t}(\mathbf{u}), \Delta} (P_i R_r)$$

Definition 3.1 (Mutual dependency). Define the binary relation *Prem* on the inductive predicate symbols of Σ as the least relation satisfying: whenever P_i occurs in the conclusion of some production in Φ , and P_j occurs amongst the premises of that production, then *Prem*(P_i, P_j) holds. Also define *Prem** to be the reflexive-transitive closure of *Prem*. Then two predicate symbols P_i and P_j are *mutually dependent* if both *Prem**(P_i, P_j) and *Prem**(P_j, P_i) hold.

Now to obtain an instance of the left-introduction rule for any inductive predicate P_j , we first associate with every inductive predicate P_i a tuple \mathbf{z}_i of k_i distinct variables (called *induction variables*), where k_i is the arity of P_i . Furthermore, we associate to every predicate P_i that is mutually dependent with P_j an arbitrary formula (called an *induction hypothesis*) F_i . Next, define the formula G_i for each $i \in \{1, \dots, n\}$ by:

$$G_i = \begin{cases} F_i & \text{if } P_i \text{ and } P_j \text{ are mutually dependent} \\ P_i \mathbf{z}_i & \text{otherwise} \end{cases}$$

We write $G_i \mathbf{t}$, where \mathbf{t} is any tuple of k_i terms, to mean $G_i[\mathbf{t}/\mathbf{z}_i]$. Then an instance of the induction rule for P_j has the following schema:

$$\frac{\text{minor premises} \quad \Gamma, F_j \mathbf{u} \vdash \Delta}{\Gamma, P_j \mathbf{u} \vdash \Delta} (\text{Ind } P_j)$$

where the premise $\Gamma, F_j \mathbf{u} \vdash \Delta$ is called the *major premise*, and for each production of Φ having in its conclusion a predicate P_i that is mutually dependent with P_j , say (Def), there is a corresponding *minor premise*:

$$\Gamma, Q_1 \mathbf{u}_1(\mathbf{x}), \dots, Q_h \mathbf{u}_h(\mathbf{x}), G_{j_1} \mathbf{t}_1(\mathbf{x}), \dots, G_{j_m} \mathbf{t}_m(\mathbf{x}) \vdash F_i \mathbf{t}(\mathbf{x}), \Delta$$

where $x \notin FV(\Gamma \cup \Delta \cup \{P_j \mathbf{u}\})$ for all $x \in \mathbf{x}$ ($FV(\cdot)$ being the usual free variable function on sets of formulas).

The induction rule for a predicate P_j can be seen to embody the natural principle of rule induction over the productions defining P_j .

Example 3.2. The induction rule for the “natural number” predicate N defined in Example 2.2 is:

$$\frac{\Gamma \vdash F0, \Delta \quad \Gamma, Fx \vdash Fsx, \Delta \quad \Gamma, Ft \vdash \Delta}{\Gamma, Nt \vdash \Delta} \text{ (Ind } N\text{)}$$

where F is the induction hypothesis associated with the predicate N . This is one way of writing the usual induction scheme for N in sequent calculus style.

Example 3.3. The induction rule (Ind E) for the “even number” predicate E defined in Example 2.2 is:

$$\frac{\Gamma \vdash F0, \Delta \quad \Gamma, Fx \vdash Hsx, \Delta \quad \Gamma, Hx \vdash Fsx, \Delta \quad \Gamma, Ft \vdash \Delta}{\Gamma, Et \vdash \Delta}$$

where F and H are the formulas associated with the (mutually dependent) predicates E and O respectively.

Definition 3.4 (Henkin validity). Let (M, \mathcal{H}) be a Henkin model for (Σ, Φ) . A sequent $\Gamma \vdash \Delta$ is said to be *true in* (M, \mathcal{H}) if for all environments ρ , whenever $M \models_\rho J$ for all $J \in \Gamma$ then $M \models_\rho K$ for some $K \in \Delta$. A sequent is said to be *Henkin valid* if it is true in all Henkin models.

Proposition 3.5 (Henkin soundness of LKID). *If there is an LKID proof of $\Gamma \vdash \Delta$ then $\Gamma \vdash \Delta$ is Henkin valid.*

Proof. A full proof appears in section 3.2 of [3]. \square

We say that a sequent $\Gamma \vdash \Delta$ is *cut-free provable* iff there is an LKID proof of $\Gamma \vdash \Delta$ that does not contain any instances of the cut or substitution rules. Our main new result about LKID is the following.

Theorem 3.6 (Cut-free Henkin completeness of LKID). *If $\Gamma \vdash \Delta$ is Henkin valid, then it is cut-free provable in LKID.*

The proof is an extension of the direct style of completeness proof for Gentzen’s LK as given in e.g. [5]. Briefly, supposing that $\Gamma \vdash \Delta$ is not cut-free provable in LKID, one uses a uniform proof-search procedure to construct a sequence of undervivable sequents $\Gamma_i \vdash \Delta_i$, which can together be used to build a syntactic countermodel to the original sequent. The required modifications in our case concern the rules for equality and inductively defined predicates, and also the need to construct a Henkin class over the model. A fully detailed proof appears in section 3.3 of [3].

Corollary 3.7 (Eliminability of cut for LKID). *If $\Gamma \vdash \Delta$ is provable in LKID then it is cut-free provable.*

Proof. If $\Gamma \vdash \Delta$ is provable in LKID, it is Henkin valid by soundness (Proposition 3.5), and hence cut-free provable in LKID by Theorem 3.6. \square

Although cut is eliminable, LKID does not enjoy the subformula property because of the induction rules. This is an unavoidable phenomenon, and corresponds to the well-known need for generalising induction hypotheses in inductive arguments. Nevertheless, cut-eliminability for LKID remains a useful property for constraining proof search; see [14] for related discussion in the intuitionistic case. Also, one can show that the eliminability of cut in LKID implies the consistency of Peano Arithmetic, so there can be no straightforward combinatorial proof of the result. A proof of this fact, together with a discussion of the connections between Corollary 3.7 and Takeuti’s Conjecture (cut-elimination for second-order sequent calculus), appears in [3], section 3.4.

4. LKID^ω: a proof system for infinite descent

We now turn to our infinitary system LKID^ω formalizing a version of proof by infinite descent. The proof rules of the system are the rules of LKID described in Section 3, except that for each inductive predicate P_i of Σ , the induction rule (Ind P_i) of LKID is replaced by the *case-split rule*:

$$\frac{\text{case distinctions}}{\Gamma, P_i \mathbf{u} \vdash \Delta} \text{ (Case } P_i\text{)}$$

where for each production having predicate P_i in its conclusion, say (Def), there is a corresponding case distinction:

$$\Gamma, \mathbf{u} = \mathbf{t}(\mathbf{x}), Q_1 \mathbf{u}_1(\mathbf{x}), \dots, Q_h \mathbf{u}_h(\mathbf{x}), \\ P_{j_1} \mathbf{t}_1(\mathbf{x}), \dots, P_{j_m} \mathbf{t}_m(\mathbf{x}) \vdash \Delta$$

where $x \notin FV(\Gamma \cup \Delta \cup \{P_i \mathbf{u}\})$ for all $x \in \mathbf{x}$. The formulas $P_{j_1} \mathbf{t}_1(\mathbf{x}), \dots, P_{j_m} \mathbf{t}_m(\mathbf{x})$ occurring in a case distinction are said to be *case-descendants* of the active formula $P_i \mathbf{u}$.

Example 4.1. The rule for N from Example 2.2 is:

$$\frac{\Gamma, t = 0 \vdash \Delta \quad \Gamma, t = sx, Nx \vdash \Delta}{\Gamma, Nt \vdash \Delta} \text{ (Case } N\text{)}$$

Example 4.2. The rule for E from Example 2.2 is:

$$\frac{\Gamma, t = 0 \vdash \Delta \quad \Gamma, t = sx, Ox \vdash \Delta}{\Gamma, Et \vdash \Delta} \text{ (Case } E\text{)}$$

Our proof system will involve infinite proofs. By a *derivation tree*, we mean a possibly infinite tree of sequents in which each parent sequent is obtained as the conclusion of an inference rule with its children as premises. We distinguish between “leaves” and “buds” in the tree. By a *leaf* we

mean an axiom, i.e. the conclusion of a 0-premise inference rule. By a *bud* we mean any sequent occurrence in the tree that is not the conclusion of a proof rule.

Definition 4.3 (LKID^ω pre-proof). An LKID^ω pre-proof of a sequent $\Gamma \vdash \Delta$ is a (possibly infinite) derivation tree \mathcal{D} , constructed according to the proof rules of LKID^ω, such that $\Gamma \vdash \Delta$ appears at the root of \mathcal{D} and \mathcal{D} has no buds.

Definition 4.4 (Trace). Let \mathcal{D} be an LKID^ω pre-proof and let $(\Gamma_i \vdash \Delta_i)$ be a path in \mathcal{D} . A *trace following* $(\Gamma_i \vdash \Delta_i)$ is a sequence (τ_i) such that, for all i :

- $\tau_i = P_{j_i} \mathbf{t}_i \in \Gamma_i$, where $j_i \in \{1, \dots, n\}$;
- if $\Gamma_i \vdash \Delta_i$ is the conclusion of (Subst) then $\tau_i = \tau_{i+1}[\theta]$, where θ is the substitution associated with the rule instance;
- if $\Gamma_i \vdash \Delta_i$ is the conclusion of (=L) with active formula $t = u$ then there is a formula F and variables x, y such that $\tau_i = F[t/x, u/y]$ and $\tau_{i+1} = F[u/x, t/y]$;
- if $\Gamma_i \vdash \Delta_i$ is the conclusion of a case-split rule then either $\tau_{i+1} = \tau_i$, or τ_i is the active formula of the rule instance and τ_{i+1} is a case-descendant of τ_i . In the latter case, i is said to be a *progress point* of the trace;
- if $\Gamma_i \vdash \Delta_i$ is the conclusion of any other rule then $\tau_{i+1} = \tau_i$.

An *infinitely progressing trace* is a trace having infinitely many progress points.

Definition 4.5 (LKID^ω proof). An LKID^ω pre-proof \mathcal{D} is an LKID^ω proof if it satisfies the *global trace condition*: for every infinite path in \mathcal{D} , there is an infinitely progressing trace following some tail of the path.

Example 4.6. Let N, E and O be the predicates given in Example 2.2. Figure 1 gives the initial part of an LKID^ω proof of the sequent $Nx_0 \vdash Ex_0, Ox_0$. The sequence $(Nx_0, Nx_1, Nx_1, Nx_1, Nx_1)$ is a trace following the displayed portion of the path in this pre-proof from the root sequent along the right-hand branch. This trace progresses because the second element Nx_1 of the trace is a case-descendant of its first element Nx_0 . One can easily see that by continuing the expansion of this derivation, we obtain an infinite tree with exactly one infinite branch. Furthermore, there is clearly a trace along this branch with infinitely many progress points: $(Nx_0, Nx_1, \dots, Nx_1, Nx_2, \dots)$, so the pre-proof thus obtained is indeed an LKID^ω proof.

The system LKID^ω is a direct extension of the cyclic proof system of [2] to infinite proof trees, and the soundness argument for LKID^ω is virtually identical to the soundness proof for the cyclic system. We briefly outline the argument, since it is helpful in understanding the global trace condition. The following lemma is a consequence of the local soundness of the proof rules.

$$\begin{array}{c}
 \text{(etc.)} \\
 \vdots \\
 \frac{}{Nx_1 \vdash Ex_1, Ox_1} \text{(Case } N) \\
 \frac{}{Nx_1 \vdash Ox_1, Osx_1} \text{(OR}_1) \\
 \frac{}{Nx_1 \vdash Esx_1, Osx_1} \text{(ER}_2) \\
 \frac{}{\vdash E0, O0} \text{(ER}_1) \\
 \frac{}{x_0 = 0 \vdash Ex_0, Ox_0} \text{(=L)} \quad \frac{}{x_0 = sx_1, Nx_1 \vdash Ex_0, Ox_0} \text{(=L)} \\
 \hline
 Nx_0 \vdash Ex_0, Ox_0 \quad \text{(Case } N)
 \end{array}$$

Figure 1. Portion of an LKID^ω proof

Lemma 4.7. Let \mathcal{D} be an LKID^ω pre-proof of $\Gamma_0 \vdash \Delta_0$, and let M be a standard model such that $\Gamma_0 \vdash \Delta_0$ is false in M under the environment ρ_0 (say). Then there is an infinite path $(\Gamma_i \vdash \Delta_i)_{i \geq 0}$ in \mathcal{D} and an infinite sequence $(\rho_i)_{i \geq 0}$ of environments such that:

1. for all i , $\Gamma_i \vdash \Delta_i$ is false in M_i under ρ_i ;
2. if there is a trace $(\tau_i = P_{j_i} \mathbf{t}_i)_{i \geq n}$ following some tail $(\Gamma_i \vdash \Delta_i)_{i \geq n}$ of $(\Gamma_i \vdash \Delta_i)_{i \geq 0}$, then the sequence $(\alpha_i)_{i \geq n}$ of ordinals defined by $\alpha_i = \text{least } \alpha \text{ s.t. } \rho_i(\mathbf{t}_i) \in P_{j_i}^\alpha$, is non-increasing. Furthermore, if j is a progress point of (τ_i) then $\alpha_{j+1} < \alpha_j$.

Proposition 4.8 (Soundness). If there is an LKID^ω proof of $\Gamma \vdash \Delta$ then $\Gamma \vdash \Delta$ is valid relative to standard models.

Proof. (Sketch) Let \mathcal{D} be an LKID^ω proof of $\Gamma \vdash \Delta$. If $\Gamma \vdash \Delta$ is not valid, i.e. false in some standard model M under some environment ρ_0 , then we can apply Lemma 4.7 to construct infinite sequences $(\Gamma_i \vdash \Delta_i)_{i \geq 0}$ and $(\rho_i)_{i \geq 0}$ satisfying properties 1. and 2. of the lemma. As $(\Gamma_i \vdash \Delta_i)_{i \geq 0}$ is a path in \mathcal{D} , there is an infinitely progressing trace following some tail of the path by Definition 4.5, so by the second property of the lemma we can construct an infinite descending chain of ordinals, which is a contradiction. \square

Note that the essential use of approximants in Lemma 4.7 means that our soundness argument only works for standard models. In fact, our main result about LKID^ω is that it is complete with respect to standard models. Indeed, we sharpen this result slightly. We say that a derivation tree is *recursive* if it is decidable whether a finite sequence of numbers corresponds to a path up the tree from the root (each number indicating the choice of rule premise determining the path) and there is a recursive function mapping finite paths in the tree to the sequents labelling their end nodes.

Theorem 4.9 (Cut-free completeness of LKID^ω). If $\Gamma \vdash \Delta$ is valid with respect to standard models of (Σ, Φ) , then it has a recursive cut-free proof in LKID^ω.

An outline proof is given in Section 5.

Although the completeness theorem shows that every valid sequent has a proof given by a recursive derivation tree, the set of valid sequents relative to standard models is non-arithmetic (one can encode true arithmetic). Thus there is no way of effectively enumerating any complete subclass of recursive proofs. Hence LKID^ω is (unsurprisingly) not suitable for formal reasoning.

The closest analogue of Theorem 4.9 we are aware of in the literature appears in [15]. There, certain *refutations* are defined, which can be seen as providing an analogous proof system to LKID^ω for Kozen's propositional μ -calculus [11]. Indeed, refutations are formulated using a trace-based proof condition very similar to Defn. 4.5. (Other similar conditions appear in [12, 19].) One of the main results of [15] is a completeness theorem for refutations. Nevertheless, the situations are quite different in many respects. The proposition μ -calculus is decidable, whereas validity (w.r.t. standard models) in our first-order logic is non-arithmetic. Also, the proof of completeness in [15] is by reduction to Borel determinacy, whereas ours, presented in Section 5, is direct.

Corollary 4.10 (Cut-eliminability for LKID^ω). *If $\Gamma \vdash \Delta$ is provable in LKID^ω then it is cut-free provable.*

Note that, unlike in LKID, cut-free proofs in LKID^ω are quite constrained: every formula appearing in a cut-free LKID^ω proof is either a subformula of a formula appearing in the root sequent, or related to such a formula by a finite number of definitional unfoldings. While not entailing a true subformula property, cut-eliminability for LKID^ω nonetheless remains close to the spirit of Girard's "purity of methods" ideal [7].

5. Outline proof of Theorem 4.9

In this section, we outline our proof (the full details of which appear in section 4.3 of [3]) of cut-free completeness for LKID^ω with respect to standard models. As in Theorem 3.6, our proof extends the direct style of completeness argument for first-order logic as in e.g. [5], but, for LKID^ω , but various additional complications arise from the need to consider infinite proofs and the global trace condition imposed upon them.

Given an arbitrary sequent $\Gamma \vdash \Delta$, we construct a recursive, possibly infinite LKID^ω derivation tree corresponding to an exhaustive search for a cut-free proof of $\Gamma \vdash \Delta$. If this tree is not an LKID^ω proof, then either it contains a bud node or there exists an infinite branch on the tree that causes the global trace condition to fail. We use the bud node or infinite branch as appropriate to construct a standard model in which $\Gamma \vdash \Delta$ is false. In the latter case, the fact that there is no infinitely progressing trace along the infinite branch is

used at two points. First, it is needed to show that no sequent on the branch is cut-free provable. Second, it is used to show that the countermodel invalidates the induced limit sequent.

Definition 5.1 (Schedule). An LKID^ω -schedule element for Σ is defined as follows:

- any formula of the form $\neg F$, $F_1 \wedge F_2$, $F_1 \vee F_2$, or $F_1 \rightarrow F_2$ is a LKID^ω -schedule element;
- for any term t of Σ , variable x , and formula F , the pairs $\langle \forall x F, t \rangle$ and $\langle \exists x F, t \rangle$ are LKID^ω -schedule elements;
- if P_i is an inductive predicate symbol of arity k_i , \mathbf{t} is a sequence of k_i terms of Σ , and r is a number such that $\Phi_{i,r} \in \Phi$, then the pair $\langle P_i \mathbf{t}, r \rangle$ is an LKID^ω -schedule element;
- for any terms t and u , variables x and y , and finite sets of formulas Γ and Δ , the tuple $\langle t = u, x, y, \Gamma, \Delta \rangle$ is an LKID^ω -schedule element.

An LKID^ω -schedule for Σ is then a recursive enumeration $(E_i)_{i \geq 0}$ of schedule elements of Σ such that every schedule element of Σ appears infinitely often in the enumeration.

Henceforth, we assume a fixed LKID^ω -schedule $(E_i)_{i \geq 0}$ and sequent $\Gamma \vdash \Delta$.

Definition 5.2 (Search tree). We define an infinite sequence of $(T_i)_{i \geq 0}$ of derivation trees such that T_0 is the single-node tree $\Gamma \vdash \Delta$ and T_i is a subtree of T_{i+1} for all $i \geq 0$. We inductively assume we have constructed T_j and show how to construct T_{j+1} .

In general T_{j+1} will be obtained by replacing certain bud nodes of T_j with derivation trees, whence it is clear that T_{j+1} is also a derivation tree as required. Firstly, we replace any bud of T_j that is an instance of the conclusion of an axiom rule with the derivation consisting of a single instance of that axiom. Let F be the formula component of E_j , the j th element in the schedule for Σ . We replace any bud of T_j that contains F with the derivation obtained by applying the sequent rule ($-L$) or ($-R$) as appropriate with active formula F , performing any required instantiations using the extra information in E_j as appropriate.

For example, if F is of the form $E_j = \langle P_i \mathbf{u}, r \rangle$, then T_{j+1} is obtained by first replacing every bud $\Gamma' \vdash \Delta'$ in T_j that satisfies $P_i \mathbf{u} \in \Gamma'$ with the derivation:

$$\frac{\text{case distinctions}}{\Gamma' \vdash \Delta'} \text{ (Case } P_i)$$

and then, assuming $\Phi_{i,r}$ is of the form (Def), and only if we have $\mathbf{u} = \mathbf{t}(\mathbf{u}')$ for some \mathbf{u}' , replacing every bud $\Gamma' \vdash \Delta'$ of the resulting tree that satisfies $P_i \mathbf{u} \in \Delta'$ with the derivation:

$$\frac{\begin{array}{l} \Gamma' \vdash Q_1 \mathbf{u}_1(\mathbf{u}'), \Delta' \quad \dots \quad \Gamma' \vdash Q_h \mathbf{u}_h(\mathbf{u}'), \Delta' \\ \Gamma' \vdash P_{j_1} \mathbf{t}_1(\mathbf{u}'), \Delta' \quad \dots \quad \Gamma' \vdash P_{j_m} \mathbf{t}_m(\mathbf{u}'), \Delta' \end{array}}{\Gamma' \vdash \Delta'} (P_i R_r)$$

Note that the above construction is performed in such way that ensures that each sequent in the tree is a subsequent of all its premises.

The *search tree* for $\Gamma \vdash \Delta$ is then defined to be T_ω , the infinite tree obtained by considering the limit as $i \rightarrow \infty$ of the sequence of (finite) derivation trees $(T_i)_{i \geq 0}$. By construction, the search tree is recursive and cut-free.

Henceforth in the proof, we assume that the search tree T_ω is not an LKID $^\omega$ proof. If T_ω is not even a pre-proof, then it contains some bud, for which we write $\Gamma_\omega \vdash \Delta_\omega$. Otherwise, T_ω is a pre-proof but not a proof. In this case, the global trace condition fails, so there exists an infinite path $\pi = (\Gamma_i \vdash \Delta_i)_{i \geq 0}$ in T_ω such that there is no infinitely progressing trace following any tail of π . We call this π the *untraceable branch* of T_ω . Define $\Gamma_\omega = \bigcup_{i \geq 0} \Gamma_i$ and $\Delta_\omega = \bigcup_{i \geq 0} \Delta_i$. (Note that we have $\Gamma_i \subseteq \Gamma_{i+1}$ and $\Delta_i \subseteq \Delta_{i+1}$ by construction of T_ω .) In either case, we call $\Gamma_\omega \vdash \Delta_\omega$ the *limit sequent*. Strictly speaking $\Gamma_\omega \vdash \Delta_\omega$ is not a sequent since the sets $\Gamma_\omega, \Delta_\omega$ may be infinite. We say that such an infinite “sequent” is *cut-free provable* to mean that some finite subsequent has a cut-free proof.

Lemma 5.3. *The sequent $\Gamma_\omega \vdash \Delta_\omega$ is not cut-free provable.*

Proof. (Sketch) The case that $\Gamma_\omega \vdash \Delta_\omega$ is a bud node is easy (if it were cut-free provable some schedule element would apply to it contradicting it being a bud node). So we assume that T_ω is a pre-proof but not a proof and let $\pi = (\Gamma_i \vdash \Delta_i)_{i \geq 0}$ be the untraceable branch. It suffices to show that no $\Gamma_i \vdash \Delta_i$ has a cut-free proof. So, for contradiction, we assume that T is a cut-free proof of $\Gamma_i \vdash \Delta_i$.

Let $\Gamma' \vdash \Delta'$ be any node in T , let (R) be the rule applied in T with active formula F (say) and conclusion $\Gamma' \vdash \Delta'$, and suppose $\Gamma' \subseteq \Gamma_j$ and $\Delta' \subseteq \Delta_j$ for some $j \geq i$. As F appears infinitely often on the schedule according to which T_ω is constructed, it follows that there is a $k \geq j$ such that F is the active formula of an instance of (R) in T_ω with conclusion $\Gamma_k \vdash \Delta_k$. Since the untraceable branch is infinite, it follows that (R) is not an axiom. Therefore, for some premise $\Gamma'' \vdash \Delta''$ of the considered instance of rule (R) in T , we have $\Gamma'' \subseteq \Gamma_{k+1}$ and $\Delta'' \subseteq \Delta_{k+1}$. This situation is illustrated in Figure 2. Since $\Gamma_i \subseteq \Gamma_{i+1}$ and $\Delta_i \subseteq \Delta_{i+1}$ for all $i \geq 0$, it follows that if (τ, τ') is a (progressing) trace following the edge $(\Gamma' \vdash \Delta', \Gamma'' \vdash \Delta'')$ in T , then $(\tau, \dots, \tau, \tau')$ is a (progressing) trace following the subpath $(\Gamma_j \vdash \Delta_j, \dots, \Gamma_k \vdash \Delta_k, \Gamma_{k+1} \vdash \Delta_{k+1})$ of π .

Now since the root of T is $\Gamma_i \vdash \Delta_i$ and trivially $\Gamma_i \subseteq \Gamma_i$ and $\Delta_i \subseteq \Delta_i$, we can repeat the argument in the preceding paragraph infinitely often to obtain a path $\pi' = (\Gamma'_j \vdash \Delta'_j)_{j \geq 0}$ in T and a sequence $k_0 < k_1 < k_2 < \dots$ of natural numbers, where $k_0 = i$, such that, for all $n \geq 0$, if (τ, τ') is a (progressing) trace following the edge $(\Gamma'_n \vdash \Delta'_n, \Gamma'_{n+1} \vdash \Delta'_{n+1})$ in T , then

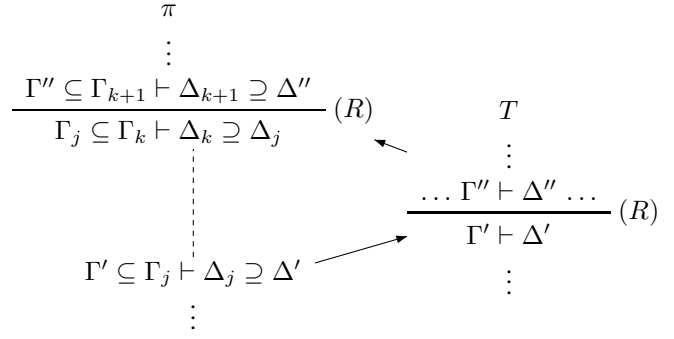


Figure 2. Part of the proof of Lemma 5.3.

$(\tau, \dots, \tau, \tau')$ is a (progressing) trace following the subpath $(\Gamma_{k_n} \vdash \Delta_{k_n}, \dots, \Gamma_{k_{n+1}} \vdash \Delta_{k_{n+1}})$ of π in T_ω .

Since T is a proof, there is an infinitely progressing trace following some tail of the constructed path π' in T . By piecing together the induced trace segments in T_ω defined above, it follows that there then is an infinitely progressing trace following some tail of the untraceable path π in T_ω . But this contradicts the defining property of π . So there cannot exist a cut-free LKID $^\omega$ proof of $\Gamma_i \vdash \Delta_i$. \square

Definition 5.4. Define the relation \sim to be the smallest congruence relation on terms of Σ that satisfies: $t_1 \sim t_2$ whenever $(t_1 = t_2) \in \Gamma_\omega$. We write $[t]$ for the equivalence class of t with respect to \sim , i.e. $[t] = \{u \mid t \sim u\}$. If $\mathbf{t} = (t_1, \dots, t_k)$ then we shall write $[\mathbf{t}]$ to mean $([t_1], \dots, [t_k])$.

Lemma 5.5. *If $t \sim u$ then, for any formula F , it holds that $\Gamma_\omega \vdash F[t/x]$ is cut-free provable if and only if $\Gamma_\omega \vdash F[u/x]$ is cut-free provable.*

Proof. By induction on the conditions defining $t \sim u$. \square

Definition 5.6 (Counter-interpretation). Define a standard model M_ω for Σ by:

- the domain of M_ω is the set of equivalence classes of Σ -terms w.r.t. \sim ;
- for any function symbol f in Σ of arity $k \geq 0$, $f^{M_\omega}([t_1], \dots, [t_k]) = [f(t_1, \dots, t_k)]$;
- for any ordinary predicate symbol Q in Σ of arity k , $Q^{M_\omega}([t_1], \dots, [t_k]) \Leftrightarrow \exists u_1, \dots, u_k. t_1 \sim u_1, \dots, t_k \sim u_k \text{ and } Q(u_1, \dots, u_k) \in \Gamma_\omega$.

(This fixes an interpretation for P_1, \dots, P_n since M_ω is a standard model.) Also, we define an environment ρ_ω for M_ω by $\rho_\omega(x) = [x]$ for all variables x . Then (M_ω, ρ_ω) is called the *counter-interpretation* for $\Gamma_\omega \vdash \Delta_\omega$.

Lemma 5.7. *For any inductive predicate P_i , if $M_\omega \models_{\rho_\omega} P_i \mathbf{t}$ then $\Gamma_\omega \vdash P_i \mathbf{t}$ is cut-free provable.*

Proof. (Sketch) It can easily be established that $\rho_\omega(\mathbf{t}) = [\mathbf{t}]$, whence we immediately have $M_\omega \models_{\rho_\omega} P_i \mathbf{t} \Leftrightarrow [\mathbf{t}] \in \bigcup_\alpha P_i^\alpha$. Now define an n -tuple of sets (X_1, \dots, X_n) by: $X_i = \{[\mathbf{t}] \mid \Gamma_\omega \vdash P_i \mathbf{t} \text{ cut-free provable}\}$ for each $i \in \{1, \dots, n\}$. By Lemma 5.5 we also have that $\Gamma_\omega \vdash P_i \mathbf{t}$ is cut-free provable iff $[\mathbf{t}] \in X_i$. It thus suffices to prove that (X_1, \dots, X_n) is a prefixed point of φ_Φ ; as the interpretation of the inductive predicates is the least prefixed point of φ_Φ , this establishes $P_i^\alpha \subseteq X_i$ and we are done.

To see that (X_1, \dots, X_n) is indeed a prefixed point of φ_Φ , it suffices to show that $\varphi_{i,r}(X_1, \dots, X_n) \subseteq X_i$ for an arbitrary production $\Phi_{i,r} \in \Phi$. This follows from the fact that cut-free provability from Γ_ω is closed under the right-introduction rule $(P_i R_r)$. \square

Lemma 5.8. *If $F \in \Gamma_\omega$ then $M_\omega \models_{\rho_\omega} F$, and if $F \in \Delta_\omega$ then $M_\omega \not\models_{\rho_\omega} F$.*

Proof. (Sketch) The proof proceeds by structural induction on F . Except for the case in which $F = P_i \mathbf{t}$ for P_i an inductive predicate, the argument is exactly as for first-order logic with equality, crucially applying Lemmas 5.3 and 5.5 to infer that atomic formulas in Δ_ω are false.

For the case $F = P_i \mathbf{t}$, the second part of the lemma follows from Lemma 5.7. For the first part, suppose for contradiction that $P_i \mathbf{u} \in \Gamma_\omega$ but $M_\omega \not\models_{\rho_\omega} P_i \mathbf{u}$. By the construction of T_ω , it follows that there is a point along the untraceable branch π at which the rule (Case P_j) is applied with active formula $P_i \mathbf{u}$, and so one of the case distinction premises of this rule instance, say:

$$\Gamma, \mathbf{u} = \mathbf{t}(\mathbf{x}), Q_1 \mathbf{u}_1(\mathbf{x}), \dots, Q_h \mathbf{u}_h(\mathbf{x}), \\ P_{j_1} \mathbf{t}_1(\mathbf{x}), \dots, P_{j_m} \mathbf{t}_m(\mathbf{x}) \vdash \Delta$$

is a subsequence of $\Gamma_\omega \vdash \Delta_\omega$. It is easy to show that the formulas $\mathbf{u} = \mathbf{t}(\mathbf{x}), Q_1 \mathbf{u}_1(\mathbf{x}), \dots, Q_h \mathbf{u}_h(\mathbf{x})$ are thus true in M_ω under ρ_ω . Since $P_i \mathbf{u}$ is false in M_ω under ρ_ω , it follows by the definition of the operator φ_Φ and the closure of $P_i^{M_\omega}$ under φ_Φ that some case-descendant of $P_i \mathbf{u}$, say $P_{j_k} \mathbf{t}_k(\mathbf{x})$, must also be false in M_ω under ρ_ω . Furthermore, there is a progressing trace $(P_i \mathbf{u}, \dots, P_i \mathbf{u}, P_{j_k} \mathbf{t}_k(\mathbf{x}))$ following a finite segment of the untraceable branch π (starting with the point where $P_i \mathbf{u}$ first appears on the left of some sequent on the branch and finishing with the case distinction in which $P_{j_k} \mathbf{t}_k(\mathbf{x})$ appears). But, since $P_{j_k} \mathbf{t}_k(\mathbf{x})$ is again false in M_ω under ρ_ω , we can apply the same argument to it as was previously applied to $P_i \mathbf{u}$ to obtain a false case-descendant of $P_{j_k} \mathbf{t}_k(\mathbf{x})$ and a progressing trace segment on π continuing from the first, and so on; and we conclude that there is an infinitely progressing trace following a tail of π , which gives the required contradiction. \square

We can now complete the proof of Theorem 4.9. Suppose that $\Gamma \vdash \Delta$ is valid, i.e. true in every standard model of (Σ, Φ) , but that the search tree T_ω for $\Gamma \vdash \Delta$ is not an

LKID $^\omega$ proof. Let $\Gamma_\omega \vdash \Delta_\omega$ be the limit sequent for $\Gamma \vdash \Delta$ with counter-interpretation (M_ω, ρ_ω) (cf. Definition 5.6). By Lemma 5.8, the sequent $\Gamma_\omega \vdash \Delta_\omega$ is false in the standard model M_ω under the environment ρ_ω . Because $\Gamma \vdash \Delta$ is a subsequence of every sequent appearing in T_ω by construction, it is a subsequence of $\Gamma_\omega \vdash \Delta_\omega$, so $\Gamma \vdash \Delta$ is false in M_ω , i.e. $\Gamma \vdash \Delta$ is invalid, which is a contradiction. Thus the recursive, cut-free search tree T_ω is a LKID $^\omega$ proof of the sequent $\Gamma \vdash \Delta$.

6. CLKID $^\omega$: a cyclic subsystem of LKID $^\omega$

In this section we revisit the “cyclic proof system” first presented in [2]. The reader is referred to [2, 3] for proofs.

In the context of the present paper, the cyclic system, here called CLKID $^\omega$, arises naturally by restricting LKID $^\omega$ to proofs given by *regular* trees, i.e. those that have only finitely many distinct subtrees. For example, although the LKID $^\omega$ proof of Figure 1 is not regular (since it contains infinitely many distinct variables x_0, x_1, x_2, \dots), it is easily transformed into a regular proof by using the substitution rule to insert a new sequent $Nx_0 \vdash Ex_0, Ox_0$ above the top-most sequent depicted. Concretely, regular LKID $^\omega$ proofs can be represented as finite graphs.

Definition 6.1 (Companion). Let B be a bud of a derivation tree \mathcal{D} . An internal node C in \mathcal{D} is said to be a *companion* for B if they have the same sequent labelling.

Definition 6.2 (Cyclic pre-proof). A CLKID $^\omega$ *pre-proof* of $\Gamma \vdash \Delta$ is a pair $(\mathcal{D}, \mathcal{R})$, where \mathcal{D} is a finite derivation tree constructed according to the rules of LKID $^\omega$ given in Section 4 and whose root is $\Gamma \vdash \Delta$, and \mathcal{R} is a function assigning a companion to every bud node in \mathcal{D} .

The *graph* of \mathcal{P} is the graph obtained from \mathcal{D} by identifying each bud node B in \mathcal{D} with its companion $\mathcal{R}(B)$.

A CLKID $^\omega$ pre-proof is then a *proof* if its graph satisfies the global trace condition of Definition 4.5. It follows immediately that any CLKID $^\omega$ proof can easily be considered as an LKID $^\omega$ proof, so that CLKID $^\omega$ is indeed a subsystem of LKID $^\omega$. Since the global trace condition is an ω -regular property, we have (as in [15, 19, 12]):

Proposition 6.3. *It is decidable whether a CLKID $^\omega$ pre-proof is a CLKID $^\omega$ proof.*

The following result from [2] gives a translation from LKID to CLKID $^\omega$:

Theorem 6.4 ([2]). *Every LKID proof of $\Gamma \vdash \Delta$ can be transformed into a CLKID $^\omega$ proof of $\Gamma \vdash \Delta$.*

Interestingly, this translation makes essential use of both the cut and substitution rules in CLKID $^\omega$. Indeed, it seems that neither rule is eliminable from the system CLKID $^\omega$. (The

importance of substitution is already illustrated in the discussion of Figure 1 above.) Nevertheless, CLKID^ω arises naturally as the restriction of a complete infinitary proof system to proofs with finite representation. The main open question relating to it is whether the converse to Theorem 6.4 holds. We strongly believe this to be the case, and hence present it as a conjecture.

Conjecture 6.5 ([2]). *If there is a CLKID^ω proof of $\Gamma \vdash \Delta$ then there is an LKID proof of $\Gamma \vdash \Delta$.*

This conjecture does not seem straightforward. For example, the methods applied in [20], which show, in a different setting, the equivalence of a weaker global proof condition with a local transfinite induction principle, do not adapt. The difficulties are reminiscent of those in proving the completeness of Kozen's axiomatization of the modal μ -calculus [11]. On the one hand, there is an analogy between regular μ -calculus *refutations* [15] and proofs in CLKID^ω . On the other hand, there is an analogy between proofs in Kozen's system, and proofs in LKID . Walukiewicz' solution to the μ -calculus completeness problem [22] established the equivalence of the two, but it is far from clear whether similar methods are applicable in our setting.

7. Future work

One direction for further research is to investigate whether more liberal subsystems of LKID^ω than CLKID^ω are also suitable for formal proof. For example, one might restrict to proofs generated by pushdown automata or by recursion schemata, over which the global trace condition is still decidable. We wonder if such proofs lead to an increase in power over regular proofs. Further, we wonder if, for some suitably chosen such class of proofs, the cut rule is eliminable. In another direction, one might also consider more restrictive systems obtained by tightening the global trace condition to improve its computational complexity (cf. [12, 19, 3]).

We comment that it should be relatively straightforward to extend our proof systems LKID , LKID^ω and CLKID^ω (together with the completeness results for the first two) to more general (co)inductive definition schemas, for example to iterated inductive definitions [13], or to the first-order μ -calculus, cf. [20].

It remains to be seen whether our investigations will be of any use to the formal reasoning community. In this regard, we believe our main contribution is in providing a firm foundation for cyclic reasoning, generalizing the heuristic conditions applied in practice. Plausibly, cyclic reasoning is likely to prove especially useful for demonstrating properties of mutually defined relations, for which the associated induction principles are often extremely complex.

References

- [1] P. Aczel. An introduction to inductive definitions. In *Handbook of Math. Logic*, pages 739–782. North-Holland, 1977.
- [2] J. Brotherston. Cyclic proofs for first-order logic with inductive definitions. In *Proceedings of TABLEAUX 2005*, Springer LNAI 3702, pages 78–92, 2005.
- [3] J. Brotherston. *Sequent Calculus Proof Systems for Inductive Definitions*. PhD thesis, Univ. Edinburgh, Nov. 2006. Available from: <http://www.doc.ic.ac.uk/~jbrother/>
- [4] A. Bundy. The automation of proof by mathematical induction. In *Handbook of Automated Reasoning*, volume I, chapter 13, pages 845–911. Elsevier, 2001.
- [5] S. R. Buss. *Handbook of Proof Theory*. Elsevier, 1998.
- [6] A. Degtyarev and A. Voronkov. Equality reasoning in sequent-based calculi. In *Handbook of Automated Reasoning*, volume I, chapter 10, pages 611–706. Elsevier, 2001.
- [7] J.-Y. Girard. *Proof Theory and Logical Complexity*, volume 1. Bibliopolis, 1987.
- [8] G. Hamilton. Póitin: Distilling theorems from conjectures. In *Proc. of the 12th Symposium on the Integration of Symbolic Computation and Mechanized Reasoning*, 2005.
- [9] J. Harrison. Inductive definitions: automation and application. In *Proc TPHOLS*, Springer LNCS 971, pages 200–213, 1995.
- [10] L. Henkin. Completeness in the theory of types. *Journal of Symbolic Logic*, 15:81–91, 1950.
- [11] D. Kozen. Results on the propositional mu-calculus. *Theoretical Computer Science*, 27:333–354, 1983.
- [12] C. S. Lee, N. D. Jones, and A. M. Ben-Amram. The size-change principle for program termination. *ACM SIGPLAN Notices*, 36(3):81–92, 2001.
- [13] P. Martin-Löf. Hauptatz for the intuitionistic theory of iterated inductive definitions. In *Proc. 2nd Scandinavian Logic Symposium*, pages 179–216. North-Holland, 1971.
- [14] R. McDowell and D. Miller. Cut-elimination for a logic with definitions and induction. *Theoretical Computer Science*, 232:91–119, 2000.
- [15] D. Niwiński and I. Walukiewicz. Games for the μ -calculus. *Theoretical Computer Science*, 163:99–116, 1997.
- [16] C. Paulin-Mohring. Inductive definitions in the system Coq: Rules and properties. In *Proceedings of TLCA'93*, Springer LNCS 664, pages 328–345, 1993.
- [17] C. Schürmann. *Automating the Meta-Theory of Deductive Systems*. PhD thesis, Carnegie-Mellon University, 2000.
- [18] K. Slind. Derivation and use of induction schemes in higher-order logic. In *Proc. TPHOLS*, Springer LNCS 1275, 1997.
- [19] C. Sprenger and M. Dam. A note on global induction mechanisms in a μ -calculus with explicit approximations. *Theoretical Informatics and Applications*, July 2003.
- [20] C. Sprenger and M. Dam. On the structure of inductive reasoning: circular and tree-shaped proofs in the μ -calculus. *Proc. FOSSACS*, Springer LNCS 2620, pp.425–440, 2003.
- [21] A. Tiu. *A Logical Framework For Reasoning About Logical Specifications*. PhD thesis, Penn. State University, 2004.
- [22] I. Walukiewicz. Completeness of Kozen's axiomatisation of the propositional μ -calculus. *Information and Computation*, 157:142–182, 2000.
- [23] C.-P. Wirth. Descente infinie + Deduction. *Logic Journal of the IGPL*, 12(1):1–96, 2004.